

Servicios RESTful

Módulo 3

Bruno Mendoza Guedes

<http://www.brunomendoza.es>

hola@brunomendoza.es

@_brunomendoza

Características Principales

Introducción

Acrónimo de Representational State Transfer.

REST es una arquitectura o estilo de arquitectura que puede ser utilizada con muchos formatos de mensajes.

Los servicios que siguen esta arquitectura se denominan servicios RESTful.

REST fue definida en los años que siguieron al 2000 por un ingeniero llamado Roy Fielding que también había contribuido al desarrollo de HTTP.

Objetivos

Los objetivos de REST incluyen:

- La *escalabilidad*. REST debe funcionar correctamente con muchos clientes y transacciones.
- Generalidad de interfaces. Debe adaptarse a cualquier proceso de negocios.
- Despliegue independiente de componentes. No es necesario utilizar sistemas operativos o lenguajes similares para que la comunicación se pueda producir. Los servicios basados en REST pueden depender de componentes intermedios para reducir la latencia, mejorar la seguridad o encapsular sistema heredados.

Mensajes de petición REST

- Los servicios RESTful esperan peticiones en forma de URIs (Uniform Resource Identifiers).
- Estas URIs pueden tener parámetros.
- El cliente tiene que formar la petición basada en la especificación del servicio.
- Las peticiones RESTful son peticiones reducidas y compactas. Esto permite que las comunicaciones sean muy rápidas y los datos devueltos pueden estar en cualquier formato.
- Recordar que SOAP utiliza muchos metadatos, una notación densa y el formato de los mensajes es único.

Mensajes de petición REST

URI o Uniform Resource Identifier

- Cadena de caracteres utilizados para identificar un recurso.
- https://en.wikipedia.org/wiki/Uniform_Resource_Identifier

URL O Uniform Resource Locator

- Es una referencia a un recurso Web que especifica su localización en una red de computadoras y un mecanismo para traerlos.
- <https://en.wikipedia.org/wiki/URL>

Mensajes de respuesta REST

- Los formatos más utilizados en servicios RESTful son XML (POX o Plain Old XML) y JSON.
- El formato XML también puede estar escrito como ATOM o RSS (Really Syndication Format).
- El formato más popular es JSON (JavaScript Object Notation).
- Los servicios RESTful también pueden enviar archivos binarios. Por ejemplo, imágenes o documentos PDF.
- Existen servicios Web que permiten al cliente escoger el formato que recibirán.
- En algunos casos es posible utilizar librerías escritas por los proveedores del servicio Web que ocultan la complejidad del sistema y peticiones asíncronas.

Acciones REST y Métodos HTTP

Introducción

REST puede ser implementando con otros protocolos diferentes de HTTP pero cuando es utilizado con HTTP es posible utilizar la estructura de sus mensajes como parte de la misma arquitectura.

Introducción

Un servicio Web RESTful implementado completamente interpretará los métodos de las peticiones HTTP como peticiones sobre alguna acción particular y los datos devueltos tendrá formato XML o JSON.

Introducción

Los métodos más comunes en un servicio Web RESTful son:

- GET,
- POST,
- PUT
- y DELETE.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

Introducción

- REST fue originalmente descrito en conjunción con HTTP.
- No todos los servicios Web utilizan todos los métodos HTTP.
- A veces todos los métodos devuelvan ejecutan la misma acción.
- Tampoco todos los cliente soportan todos los métodos HTTP.

Introducción

- REST fue originalmente descrito en conjunción con HTTP.
- No todos los servicios Web utilizan todos los métodos HTTP.
- A veces todos los métodos devuelvan ejecutan la misma acción.
- Tampoco todos los cliente soportan todos los métodos HTTP.

Peticiones RESTful

Peticiones

- La sintaxis entre dos servicios Web diferentes puede variar mucho.
- La sintaxis depende del proveedor del servicio Web.
- Es común que los servicios Web gestionen peticiones sólo GET y POST.
Sobre todo en servicios de sólo lectura.
- Además de conocer los métodos que soporta el servicio Web es necesario conocer también qué atributos soporta.
- Estos pueden ser parámetros de tipo query o parámetros embebidos. Estos últimos son parte de un mensaje que utiliza el método POST u otro método.

Ejemplo de petición POST

```
POST /rest/algun-servicio.php HTTP/1.1
```

```
Host: cualquier.servicio.web.com
```

```
Accept: application/xml
```

```
Content-Type: application/xml
```

```
Content-Length: 100
```

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<parameters>
```

```
  <packageid>5</packageid>
```

```
</parameters>
```


Peticiones

- **Content-Type** indica que el formato de envío y **Accept** indica el formato que deseamos de vuelta.
- La manera de programar una petición es completamente diferente dependiendo del lenguaje y la plataforma que hayamos escogido.
- La mayor parte de los lenguajes de programación contienen herramientas incorporadas (built-in) para realizar peticiones.
- Y muchas plataformas contienen librerías que ofrecen interfaces para gestionar las peticiones escondiendo parte de la complejidad del uso del servicio.

Peticiones: Java

Paquetes o clases Java

- HttpURLConnection
 - <https://docs.oracle.com/javase/8/docs/api/java/net/URLConnection.html>
- HttpClient de Apache.
 - <https://hc.apache.org/httpcomponents-client-ga/>

Peticiones: Android

Librerías para Android

- Volley
 - <https://developer.android.com/training/volley/index.html>
- OkHttp
 - <http://square.github.io/okhttp/>
- Retrofit
 - <http://square.github.io/retrofit/>

Peticiones: JavaScript

En Javascript

- XMLHttpRequest object
 - <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>
 - https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest#Browser_compatibility
- JQuery AJAX Request
 - <http://api.jquery.com/category/ajax/>

Respuestas RESTful

Respuestas: formatos

La arquitectura no especifica ningún formato de respuesta en particular, pero la mayoría utiliza JSON o XML (POX, ATOM o RSS).

Respuestas: RSS

RSS 1.0 o RDF Site Summary.

RSS 2.0 o Really Simple Syndication. Open Source. No existe especificación.

Puede ser utilizado para otras estructura de datos diferentes a la estructura de datos para la que fue diseñado.



Respuestas: RSS

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
  <title>RSS Title</title>
  <description>This is an example of an RSS feed</description>
  <link>http://www.example.com/main.html</link>

  <!-- Original Publication Date -->
  <lastBuildDate>Mon, 06 Sep 2010 00:01:00 +0000 </lastBuildDate>

  <!-- Last Update Date -->
  <pubDate>Sun, 06 Sep 2009 16:20:00 +0000</pubDate>

  <!-- Lifetime of data -->
  <ttl>1800</ttl>

  <item>
    <title>Example entry</title>
    <description>Here is some text containing an interesting description.</description>
    <link>http://www.example.com/blog/post/1</link>
    <guid isPermaLink="false">7bd204c6-1655-4c27-ae4e-53f933c5395f</guid>
    <pubDate>Sun, 06 Sep 2009 16:20:00 +0000</pubDate>
  </item>

</channel>
</rss>
```


Respuestas: RSS

Otro ejemplo:

<https://groups.google.com/forum/#!aboutgroup/mozilla.dev.tech.xml>

Respuestas: ATOM

- A veces se utiliza para referirse a un protocolo de publicación.
- Un protocolo basado en HTTP para gestionar recursos Web.
- En el contexto donde nos encontramos nos referimos a **ATOM Syndication Format**.
- Otro dialecto XML para alimentar la Web y como contenedor de otros tipos de datos.
- Es el formato utilizado por la arquitectura OData y permite manejar más tipos de datos que RSS.
- Especificación:
 - Internet Engineering Task Force: <https://tools.ietf.org/html/rfc4287>

Respuestas: ATOM

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>Example Feed</title>
  <subtitle>A subtitle.</subtitle>
  <link href="http://example.org/feed/" rel="self" />
  <link href="http://example.org/" />
  <id>urn:uuid:60a76c80-d399-11d9-b91C-0003939e0af6</id>
  <updated>2003-12-13T18:30:02Z</updated>
  <entry>
    <title>Atom-Powered Robots Run Amok</title>
    <link href="http://example.org/2003/12/13/atom03" />
    <link rel="alternate" type="text/html" href="http://example.org/2003/12/13/atom03.html"/>
    <link rel="edit" href="http://example.org/2003/12/13/atom03/edit"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2003-12-13T18:30:02Z</updated>
    <summary>Some text.</summary>
    <content type="xhtml">
      <div xmlns="http://www.w3.org/1999/xhtml">
        <p>This is the entry content.</p>
      </div>
    </content>
    <author>
      <name>John Doe</name>
      <email>johndoe@example.com</email>
    </author>
  </entry>
</feed>
```

Respuestas: POX

- Utilizan sus propios esquema o dialecto.
- Estos esquemas definen la estructura de datos subyacente.
- Estos esquemas pueden ser un estándar definido entre sistemas o único para un servicio Web.
- <http://www.carimagery.com/api.asmx> (SOAP mover)

Respuestas: JSON

- Acrónimo de Javascript Object Notation.
- Creado por una empresa llamada Software State en 2001.
- Popularizado a través del libro 'JavaScript, The Good Parts' por Douglas Crawford.
- Utiliza la notación de JavaScript para codificar datos estructurados.
- Es un formato compacto consume menos ancho de banda, menos memoria y menos recursos para su procesamiento.

Respuestas: JSON

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```